Partial Derivative Computation for Bounded-Impulse Trajectory Models Using Two-Sided Direct Shooting. Part 2: Application

Donald H. Ellison [*] and Bruce A. Conway [†]

University of Illinois at Urbana-Champaign, Urbana, IL, 61801, USA

Jacob A. Englander [‡]

NASA Goddard Space Flight Center, Greenbelt, MD, 20771, USA

Martin T. Ozimek [§]

Johns Hopkins University Applied Physics Laboratory, Laurel, MD, 20723, USA

In the companion paper, analytic methods were presented for computing the Jacobian entries for two-sided direct shooting trajectory models that utilize the bounded-impulse approximation. In this paper we discuss practical implementation considerations. Efficient computation of the mathematical components required to compute the partials is discussed and a guiding numerical example is provided for validation purposes. A solar electric power model suitable for preliminary mission design is presented, including a method for handling thruster cut-off events that result in non-smooth derivatives. The challenges associated with incorporating the SPICE ephemeris system into an optimization framework are discussed and an alternative is presented that results in smooth time partials. Application problems illustrate the benefits of employing analytic Jacobian calculations vs. using the method of finite differences. The importance of accurately modeling hardware and operational constraints at the preliminary design stage, and the benefits of using an analytic Jacobian in a solver that combines the monotonic basin hopping heuristic method with a local gradient search are also explored.

[1]Ph. D. Candidate, Department of Aerospace Engineering, 104 South Wright Street, Urbana, IL, Mail Code-236, Student Member AIAA
[2]Professor Emeritus, Department of Aerospace Engineering, 104 South Wright Street, Urbana, IL, Mail Code-236, Associate Fellow AIAA
[3]Aerospace Engineer, Navigation and Mission Design Branch, 8800 Greenbelt Rd, Greenbelt, MD 20771, Member AIAA
[4]Mission Design Engineer, 11100 Johns Hopkins Road, Laurel, MD, Member AIAA, AAS

Nomenclature

α	=	logistics function sharpness
$\mathbf{c}$	=	nonlinear program constraint vector
δ_{power}	=	spacecraft power margin
h	=	finite differencing step size
$\mathbf{M}$	=	maneuver transition matrix
m	=	spacecraft mass
N_{active}	=	number of active thrusters
$\mathcal{N}(\mu, \sigma)$	=	normal distribution with mean μ
		and standard deviation σ
P	=	power available to the solar electric propulsion unit
P_0	=	power generated by the solar electric array at 1 A.U.
$P_{0-\text{BOL}}$	=	power generated by the solar electric array at 1 A.U.
		at beginning of life
P_{eff}	=	effective power available to a single electric thruster
$P_{\text{generated}}$	=	total power generated by the solar electric array
$P_{\text{s/c}}$	=	power required to operate the spacecraft bus
$\mathbf{r}$	=	spacecraft position vector w.r.t. central body
T	=	thrust
t	=	current epoch
τ	=	solar cell decay rate
$\mathbf{u}$	=	control vector
$\mathbf{v}$	=	spacecraft velocity vector w.r.t. central body
$\mathbf{x}$	=	nonlinear program decision vector
$\mathbf{X}$	=	spacecraft state vector
Δt	=	propagation time of a trajectory segment
Δt_{flight}	=	mission time-of-flight
Δt_p	=	phase time-of-flight

$\boldsymbol{\Theta}$ = time variable connection matrix

$\boldsymbol{\Phi}$ = state transition matrix

$\dot{(\cdot)}$ = first time derivative

$(\cdot)^{\dagger}$ = phase match point

$(\cdot)_0$ = initial epoch

$(\cdot)_f$ = end of a phase

$(\cdot)_{\max}$ = maximum value

$(\cdot)_{\min}$ = minimum value

$(\cdot)_{\odot}$ = solar quantity measured in the frame of the central body

I. Introduction

The bounded impulse approximation for spacecraft trajectories, in both high and low-thrust regimes, has an important role in many preliminary trajectory design techniques [1–5]. Mission design efforts have benefited greatly from the execution speed of software implementations of these models, and they are valued for their ability to rapidly generate medium-fidelity solutions capable of steering trade studies and proposal efforts [6, 7].

The analytic techniques that were the focus of the companion paper were developed with the multiple gravity assist low-thrust (MGALT) and multiple gravity assist with n deep space maneuvers using shooting (MGAnDSMs) transcriptions in mind, but can be extended to any bounded impulse trajectory model [8]. The optimization of the trajectories encoded with these transcriptions is typically performed with a nonlinear programming (NLP) solver (such as SNOPT [9] or IPOPT [10]), and one of the critical pieces of information required by such a tool are the partial derivatives obtained by taking the partial derivative of the constraint vector $\mathbf{c}$ with respect to the vector of decision variables $\mathbf{x}$ that constitute the problem to be solved:

$$\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \tag{1}$$

The vectors $\mathbf{c}$ and $\mathbf{x}$ are typically comprised of tens to hundreds of entries for most most trajectory models, potentially resulting in a large number of partial derivatives that must be computed even for sparse problems.

The matrix in Eq. (1), referred to as the Jacobian, can be calculated using a variety of techniques, such as finite differencing, however there are many advantages to determining it analytically. The companion paper developed the theory required to do that for bounded impulse transcriptions. The software infrastructure to efficiently implement these analytic techniques can be tedious to establish, so the first part of this paper provides what are hoped to be some helpful guidelines and suggestions to facilitate such an effort.

The next part of this paper describes techniques for modeling spacecraft solar electric power systems that meet the standards for a NASA Discovery mission proposal. This includes a discussion of the impact that modeling discrete

thruster activation events has on a numerical optimizer such as SNOPT. The problems associated with incorporating the ephemeris system SPICE into the optimization process are also discussed and an alternative is proposed that is used throughout this work.

Finally, four application problems that underscore the benefits gained from utilizing an analytically computed Jacobian in the optimizer will be presented. Several of these examples also investigate the effects that using these analytic techniques has on a solver scheme that incorporates both local gradient searches using an NLP solver as well as a stochastic search component realized with the monotonic basin hopping (MBH) heuristic algorithm [11–14].

II. State and Derivative Propagation Sweeps

Section III. A. of the companion paper discusses a general method for mapping derivative information along a two-point shooting phase towards the defect point. To summarize, this is achieved via the alternating multiplication of linear matrix maps across propagation arcs and bounded impulse maneuver events. An augmented version of the two-body perturbation state transition matrices (STMs) maps first order variations in the spacecraft state across Keplerian arcs and similarly maneuver transition matrices (MTMs) map first order variations across impulsive maneuvers. These matrices may be chain-multiplied due to their transitive nature in order to propagate derivative information from any point along the phase to the match point. This is accomplished in three steps as shown in Figure 1.

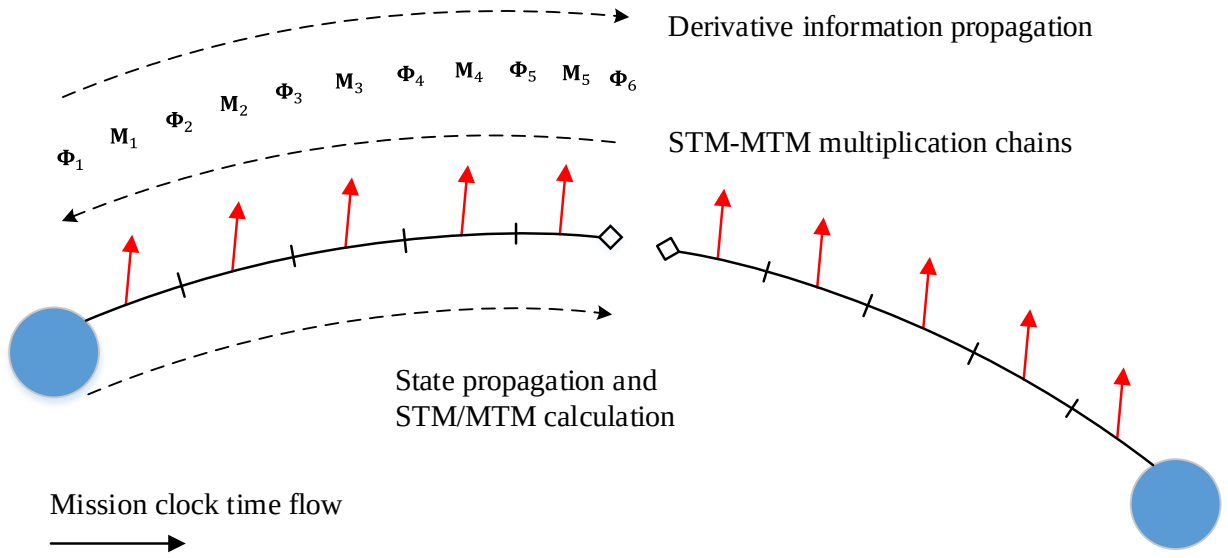

Figure 1. Example 10 segment MGALT phase. Match point derivatives are calculated in three steps 1) the STMs and MTMs are computed as the state is propagated to the match point 2) the STM-MTM multiplication chains are assembled from the match point outwards to the phase boundaries 3) derivative information is propagated from points along the phase to the match point.

First, the spacecraft state ($\mathbf{X}$) is propagated from both phase boundaries inward to the defect in the center of the

phase. As this propagation proceeds, the STMs and MTMs are calculated and stored. Then STM-MTM matrix chains are constructed in a backwards sweep that opposes the direction of state propagation. It is most efficient to perform this sweep beginning at the match point and then proceeding towards the phase boundary as the STM-MTM chains can be successively constructed by recursively combining previously computed sub-chains. This idea is illustrated in Figure 2 for a forward half-phase. The same holds true for backward propagated half-phases.

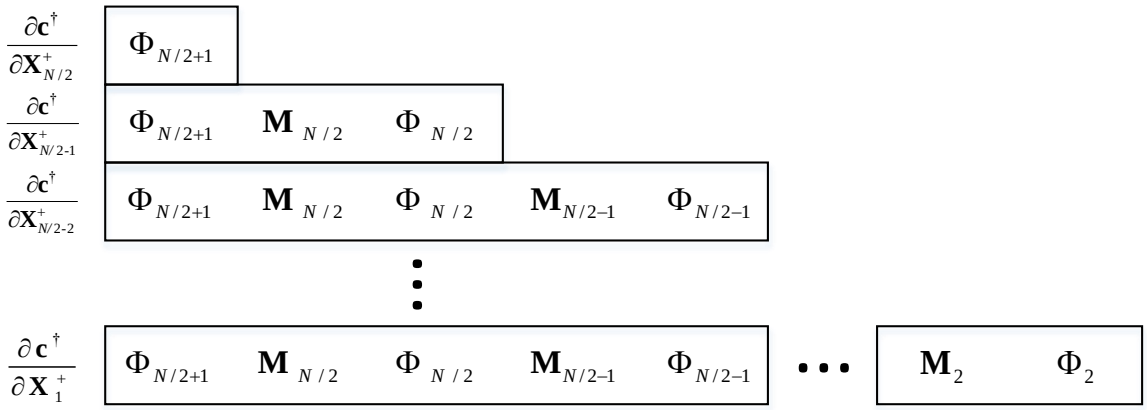

Figure 2. Recursive construction of the STM-MTM chain for a forward propagated half-phase.

Finally, the STM-MTM chains can be used to transmit derivative information from any point along the phase to the match point in order to compute the defect constraint $\mathbf{c}^\dagger$ there. The calculation of STMs and MTMs and the construction of the matrix chains represents, by far, the majority of the computational effort required to evaluate a two-point shooting phase as depicted in Figure 1. In order to assist implementation, a sample numerical STM-MTM chain multiplication is provided in Appendix (D), which makes use of the analytic expressions presented in the companion paper [8] as well as the hardware modeling that will be discussed in section III of this paper.

III. Spacecraft Power System Modeling

If the spacecraft is using a solar electric propulsion (SEP) system, the maximum amount of thrust that the system can produce depends on the power available, which depends on its distance from the sun. This must be modeled in the low-thrust global optimization even at the preliminary design stage. In this work we will focus on models of real thruster hardware. Typically trajectory design engineers are supplied with polynomial representations of the thrust and mass flow rate of a given thruster, *e.g.*,

$$T = e_T P_{\text{eff}}^4 + d_T P_{\text{eff}}^3 + c_T P_{\text{eff}}^2 + b_T P_{\text{eff}} + a_T \tag{2}$$

$$\dot{m}_{\text{max}} = e_m P_{\text{eff}}^4 + d_m P_{\text{eff}}^3 + c_m P_{\text{eff}}^2 + b_m P_{\text{eff}} + a_m \tag{3}$$

where P_{eff} is the power available to each thruster,

$$P_{\text{eff}} = P/N_{\text{active}} \tag{4}$$

and N_{active} is the number of thrusters firing at any point in time. The polynomial coefficients used in Eq. (2) and (3) are provided in Table 10 in Appendix.

Equations (2) and (3) are valid over a range $[P_{\text{min}}, P_{\text{max}}]$ where P_{min} represents the minimum amount of power necessary to turn on the thruster's power processing unit (PPU) at the lowest setting and P_{max} represents the maximum amount of power that the PPU can safely accommodate. The total power available to the electric propulsion system (P) is the difference between the power generated by the spacecraft $P_{\text{generated}}$ and the power required to operate the spacecraft bus $P_{\text{s/c}}$,

$$P = (1 - \delta_{\text{power}})\left(P_{\text{generated}} - P_{\text{s/c}}\right) \tag{5}$$

where δ_{power} is the propulsion power margin. In this work, the power delivered by a solar array is given by [15]:

$$P_{\text{generated}} = \frac{P_0}{r_{s/\odot}^2}\left(\frac{\gamma_0 + \gamma_1/r_{s/\odot} + \gamma_2/r_{s/\odot}^2}{1 + \gamma_3 r_{s/\odot} + \gamma_4 r_{s/\odot}^2}\right) \tag{6}$$

where the γ_i are solar panel coefficients typically determined from experimental data, $r_{s/\odot}$ is the distance between the sun and the spacecraft in Astronomical Units (AU) and P_0 is the "base power" delivered by the array at 1 AU. P_0 is in turn a function of the time since launch,

$$P_0 = P_{\text{0-BOL}}\,(1 - \tau)^t \tag{7}$$

where $P_{\text{0-BOL}}$ is the base power delivered by the array at 1 AU on the day of launch, τ is the decay rate of the solar arrays measured as a percentage per year, and t is the time since launch in years. Equation (7) may also be used to model the decay of an radioisotope thermal generator (RTG) or advanced Stirling radiosotope generator (ASRG) power system.

The power required by the spacecraft bus $P_{\text{s/c}}$ is also modeled as a polynomial,

$$P_{\text{s/c}} = a_{\text{s/c}} + b_{\text{s/c}}/r_{s/\odot} + c_{\text{s/c}}/r_{s/\odot}^2 \tag{8}$$

where $a_{\text{s/c}}$, $b_{\text{s/c}}$, and $c_{\text{s/c}}$ are specified by the mission designer.

The most interesting case is when $P_{\text{eff}} > P_{\text{max}}$ or $P_{\text{eff}} < P_{\text{min}}$, and therefore thrusters must be switched on or off. If $P_{\text{eff}} > P_{\text{max}}$ then either an additional thruster must be switched on, or if no other thrusters are available, P_{eff} must be clipped to P_{max}. If $P_{\text{eff}} < P_{\text{min}}$ then a thruster must be switched off. Indeed, for propulsion modules featuring multiple thrusters, several different thruster logic programs may be followed. These include, but are not limited to activating the maximum number of thrusters for a given P, activating the minimum number of thrusters for a given P, maximize

the total propulsion efficiency for a given P, maximize the total thrust for a given P and maximize the total specific impulse for a given P. Two of these thruster logic programs are provided in Appendix B. A discontinuity exists in Equations (2) and (3) at the boundaries where $P_{\text{eff}} = P_{\text{max}}$ or $P_{\text{eff}} = P_{\text{min}}$. This discontinuity is very confusing to gradient-based optimizers, especially in the case where there is not enough power to turn on any thrusters at all. It is desirable to smooth the power and propulsion models and remove the discontinuity. McConaghy [16] proposed smoothing the propulsion model using the "smoothstep" technique from the field of computer graphics. However we have developed a different approach for this work.

Heaviside [17] defined the unit step function as instantaneously taking half value at the point of transition. It is then possible to approximate the step function using the logistics function,

$$H(x) = \lim_{\alpha \to \infty} \frac{1}{1 + \exp(-2\alpha x)}. \tag{9}$$

Equation (9) is continuously differentiable and therefore eliminates the problems that a gradient-based solver would have with a step function. In the context of multi-thruster switching, we define a set of Heaviside step functions $H_i(P)$,

$$H_i(P) = \frac{1}{1 + \exp(-2\alpha(P - P_i^*))} \tag{10}$$

where each Heaviside step function $H_i(P)$ defines the switch state of the i^{th} thruster and α defines the sharpness of the transition. The larger the value of α, the closer $H_i(P)$ approximates the Heaviside step function. However, while the derivatives $H_i'(P)$ increase as α increases, they remain finite and $H_i(P)$ remains continuous. We find that the optimizer behaves best when the derivatives are reasonably small (*i.e.* small α) but we also find that if α is too small then the thruster model will frequently report a fractional non-integer $H_i(P)$. N_{active} may then be defined as,

$$N_{\text{active}} = \sum_{i=1}^{N} H_i(P) \tag{11}$$

To fully define N_{active}, it is necessary to define the transition powers P_i at which a thruster would be switched on and off. In this work it is assumed that as few thrusters as possible are used for a given available power and so each P_i is an integer multiple of P_{max} except for P_1, which is equal to P_{min}. Alternatively if as many thrusters as possible are activated, then each P_i would be an integer multiple of P_{min}. It is also possible to define other switching laws such as maximum thrust or maximum specific impulse (I_{sp}), in which case one would need to compute the P_i where those merit functions change as a function of number of thrusters.

Figure 3 shows the available thrust for a notional system with four BPT-4000 thrusters, each with $P_{\text{min}} = 0.302$ kW. The plot is focused on the region between 2 and 4 AU where the thruster transitions occur. The spacecraft's solar array can provide 10 kW at 1 AU and the spacecraft bus requires 0.5 kW at all times. Several values of α are

shown, each a different compromise between smoothness and accuracy. One can see that the green line, representing $\alpha = 1000$, closely approximates the unsmoothed black line that does not have continuous derivatives. However the smoothing method described here removes that discontinuity and significantly improves the robustness of the solver. A value of $\alpha = 100$ is recommended as a good compromise between well-behaved derivatives and accuracy.

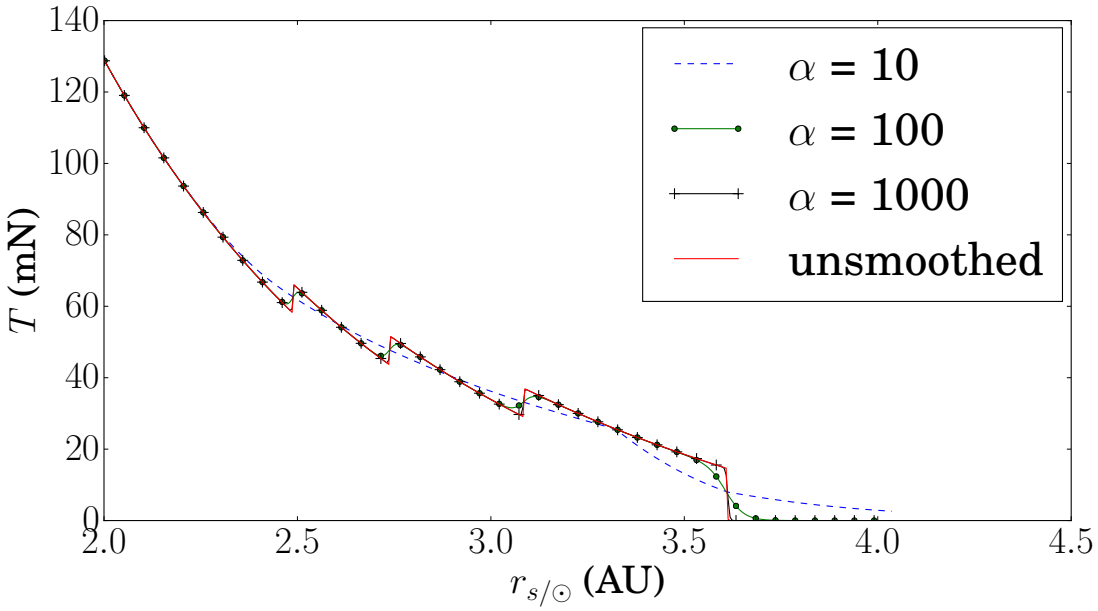

Figure 3. Thrust vs. distance from sun for various values of α. The propulsion system consists of 4xBPT-4000 thrusters with $P_{BOL} = 10$ kW.

IV. Partial Derivatives of the Ephemeris

In most trajectory design applications, ephemeris data for solar system bodies is provided using the SPICE ephemeris toolkit [18]. SPICE provides a compact, high-precision approximation to the actual integrated ephemerides by using Chebychev polynomials and is considered industry standard for astronomers and planetary scientists, and also for most trajectory design work. However, SPICE suffers from two major limitations that make it less than ideal for optimization. First, in the interest of maintaining a small memory footprint, SPICE reads ephemeris data directly from the hard drive and is therefore quite slow. This was a very helpful feature in the past when computer memory was limited, but it also is a handicap in trajectory optimization. Second, and very specific to this work, SPICE does not provide derivatives of the body states, nor does it provide access to the underlying polynomial that might be used to construct derivatives. Furthermore, in the interest of maintaining accuracy, each of the six states (x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$) is fit to a separate polynomial for most SPICE ephemeris types. Velocity in SPICE is therefore not the derivative of position. As a result of these limitations, SPICE is not the ideal ephemeris package for trajectory optimization because it does not work with an analytical derivatives formulation such as that described in this work.

Several attempts were made to circumvent these issues. The first approach was to simply compute all derivatives with respect to time using finite differencing. This was slow and had poor convergence properties. The second approach was to finite difference just the ephemeris itself and then chain the resulting boundary state derivative with the analytical derivatives derived in this work. The second approach was much faster but delivered a poor approximation to the actual derivatives and led to poor convergence. The third approach was to fit a cubic spline to the ephemeris and cache the spline coefficients in RAM. This approach was first proposed by Arora and Russell [19]. The approach used in this work is similar to [19] except that where Arora and Russell created ephemeris archives and saved them to the hard drive for later re-use, the SplineEphem package created for this work performs the spline fitting at bootstrap of the optimization program. This approach is simpler to use at the cost of a few seconds of load time prior to each optimization run.

IV. A. SplineEphem

The creation of a SplineEphem fit for a given solar system body consists of two steps. First, SPICE is used to generate set of ephemeris points over which the spline will be fit. The spline fits themselves are then computed using the GNU Scientific Library (GSL) spline package [20]. Figures 4 and 5 show the error in position and velocity for a SplineEphem fit to Jupiter over 1000 days, with 100 ephemeris points per orbital period of Jupiter. The maximum position error in this example is 414 km and the maximum velocity error is 4.3 m/s. This is more than sufficient for the preliminary trajectory design studies in this work, but should better accuracy be required for high-fidelity planning, the number of sample points may be increased at the cost of memory footprint.

In addition to providing analytical derivatives that themselves lead to improved speed and convergence, SplineEphem is much faster than SPICE. Speed was tested using 1e+7 samples. If the samples are sequential in time then SplineEphem is 12 times faster than SPICE. If the sample epochs are randomized, then SplineEphem is 82 times faster.

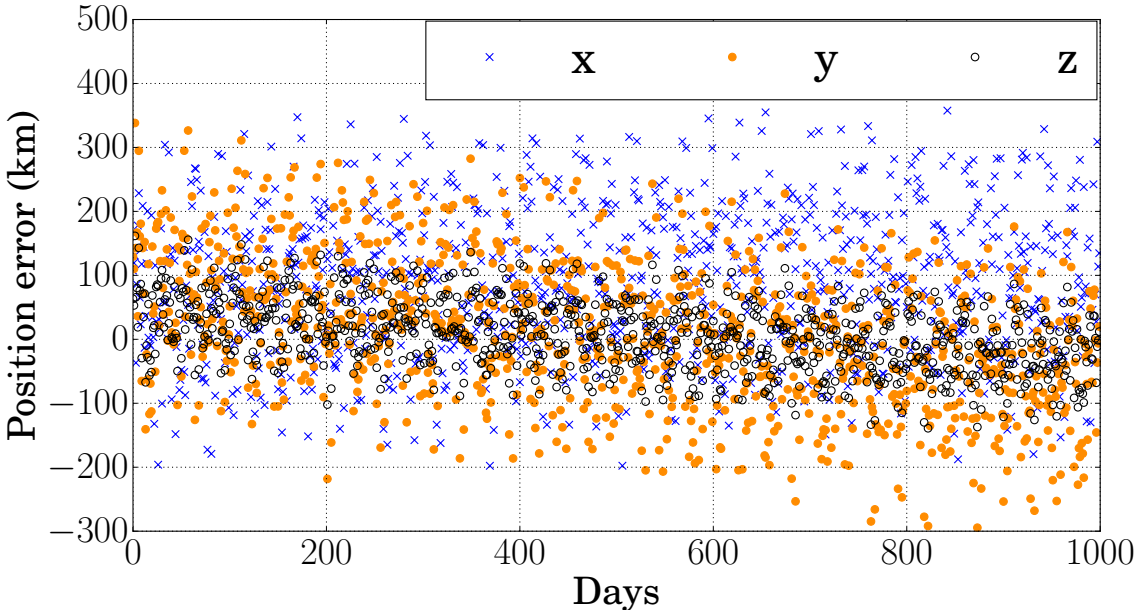

Figure 4. SplineEphem error relative to SPICE for the position of Jupiter relative to the Sun, over a 1000-day period.

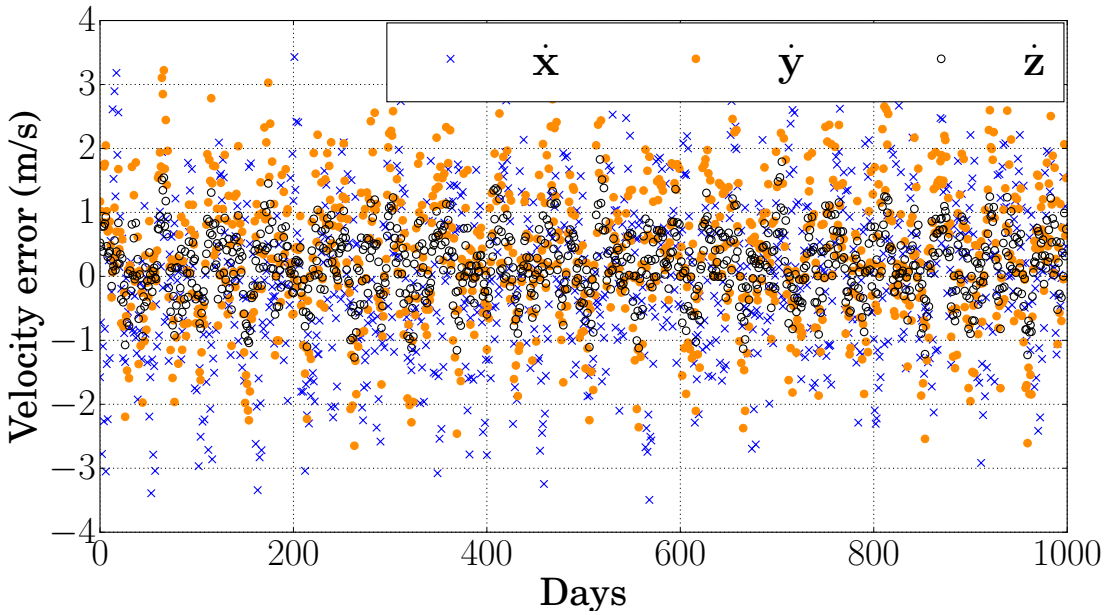

Figure 5. SplineEphem error relative to SPICE for the velocity of Jupiter relative to the Sun, over a 1000-day period.

V. Applications

Four problems were selected to showcase the benefits of using an analytically calculated Jacobian vs. calculating it using the method of finite differences. The first three problems make use of the continuous-thrust MGALT model

and the fourth uses the high-thrust MGAnDSMs transcription.

The first problem is a notional mission to Uranus that features accurate launch vehicle and solar electric power system models. The purpose of this problem is to test the convergence characteristics of the nonlinear programming solver SNOPT in the context of using analytic vs. finite differenced Jacobian calculations. In addition, solution details showing thruster switching, throttle settings as well as spacecraft power generation and consumption provide a compelling argument for the necessity of modeling these details in a preliminary design setting.

The second application problem is a solar electric Mercury rendezvous mission that was described by Sauer [21] and Debban et al. [22]. For this problem, the effect of derivative accuracy on the NLP solver's convergence is explored in further detail. A series of experiments are performed where a random vector is added to the optimal decision vector, then the problem is re-solved once with analytic derivatives and once with finite differencing. The convergence properties of SNOPT using the two partial derivative computation methods is compared.

The third problem simulates a mission to the asteroid Odysseus and introduces a complex operational constraint that pushes the limits of what the solver is able to achieve a feasible solution for, let alone an optimal one. This problem was solved with the monotonic basin hopping (MBH) heuristic method combined with a local gradient search provided by SNOPT. The MBH method randomly initializes the problem's decision variables and then explores the problem's design space using stochastic operations as well as local gradient searches. The problem was solved numerous times using our analytic techniques, finite differencing, and finally with a "two-phase" strategy that combines both methods for providing the derivative computations required by SNOPT.

The final problem is a recreation of the Cassini mission to Saturn and uses the high-thrust model MGAnDSMs. The MBH+SNOPT scheme was also used to solve this problem.

All application problems in this work were solved using the Taub computing cluster at the University of Illinois at Urbana-Champaign. The compute nodes in this cluster are Intel HP X5650 2.66 GHz 6-core processors.

V. A. Application to Uranus Probe

This problem is inspired by the Ice Giants Decadal Mission Study Final Report from the NASA Planetary Science Decadal Survey carried out by the Applied Physics Laboratory [23]. The thruster mass flow rate and thrust polynomial coefficients, sufficiently accurate for preliminary design purposes, are from the 2014 Discovery proposal NEXT thruster guide [24] and are provided in the Appendix (Table 10). The problem was first solved assuming all parameters in Table 1 [25] except for the following: a minimum earth flyby altitude of 300 km, a maximum time of flight of 13.5 years, a simplified solar array model $\{\gamma_0 = 1.0, \gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = 0.0\}$ and no forced coasts. This solution was then used as an initial guess for the problem as described in Table 1. The NLP solver was run using a finite differenced Jacobian and then using an analytic Jacobian.

The results summarized in Table 2 indicate that while both Jacobian calculation methods resulted in essentially equal final cost values, the NLP solver was able to achieve a significantly tighter feasibility tolerance in fewer iterations using

Table 1. Uranus Probe mission parameters

Parameter	Value
Maximum allowed initial mass	4200 kg
Earliest allowed launch date	January 1st, 2020
Launch window	365.25 days
Maximum flight time	13 years
Launch vehicle	Atlas V (531) NLS-1
Launch C3	11.834 km^2/s^2
Launch declination bounds	$[-28.5^\circ, 28.5^\circ]$
Arrival type	intercept
Maximum arrival v_∞	7.21 km/s
Thruster	2 x NEXT TT10 high I_{sp}
Throttle logic	minimum number of thrusters
Thruster duty cycle	0.9
Solar power coefficients	$\gamma_0 = 1.1705$ $\gamma_1 = 0.0289$ $\gamma_2 = -0.2197$ $\gamma_3 = -0.0202$ $\gamma_4 = -0.0001$
Spacecraft bus power requirement coefficients	$a_{s/c} = 0.8$ $b_{s/c} = 0.0$ $c_{s/c} = 0.0$
Solar array BOL power (launch at 1 A.U.)	26.0 kW
Power margin (δ_{power})	15%
Flyby sequence	E-E-U
Minimum earth flyby altitude	1000 km
Post-launch checkout coast	30 days
Pre-Earth flyby coast	42 days
Post-Earth flyby coast	7 days
Number of segments per phase	100
Ephemeris	SplineEphem
SNOPT feasibility tolerance	1.0e-5
SNOPT optimality tolerance	1.0e-6
Objective function	maximize final mass
SNOPT max. runtime	3600 s
SNOPT major iteration limit	8000

Table 2. SNOPT convergence statistics for the Uranus Probe mission

Metric	Analytic	Finite Differences
Optimal normalized cost	0.80964301784	0.80964033542
SNOPT major iterations	755	1162
Feasibility achieved	1.07094e-13	2.62403e-09
SNOPT execution time (s)	15	112
Optimality tolerance met?	Yes	No

analytic gradients as opposed to finite differenced ones. The analytic Jacobian also led to an optimal exit condition for the NLP solver.

Figure 8 shows the throttle setting used as well as the number of active thrusters throughout the mission. The thruster count transitions were smoothed using the technique described in section III. Figure 9 depicts the power generated by the spacecraft's solar array, the power utilized by the thrusters and the thrust achieved. It is important to note that while a bang-bang control structure is exhibited in Figure 8, the power model does not allow for the thrusters to operate at their maximum input voltage setting at all times. The gap present between the two power curves in Figure 9 is due to the power margin and spacecraft bus power requirements.

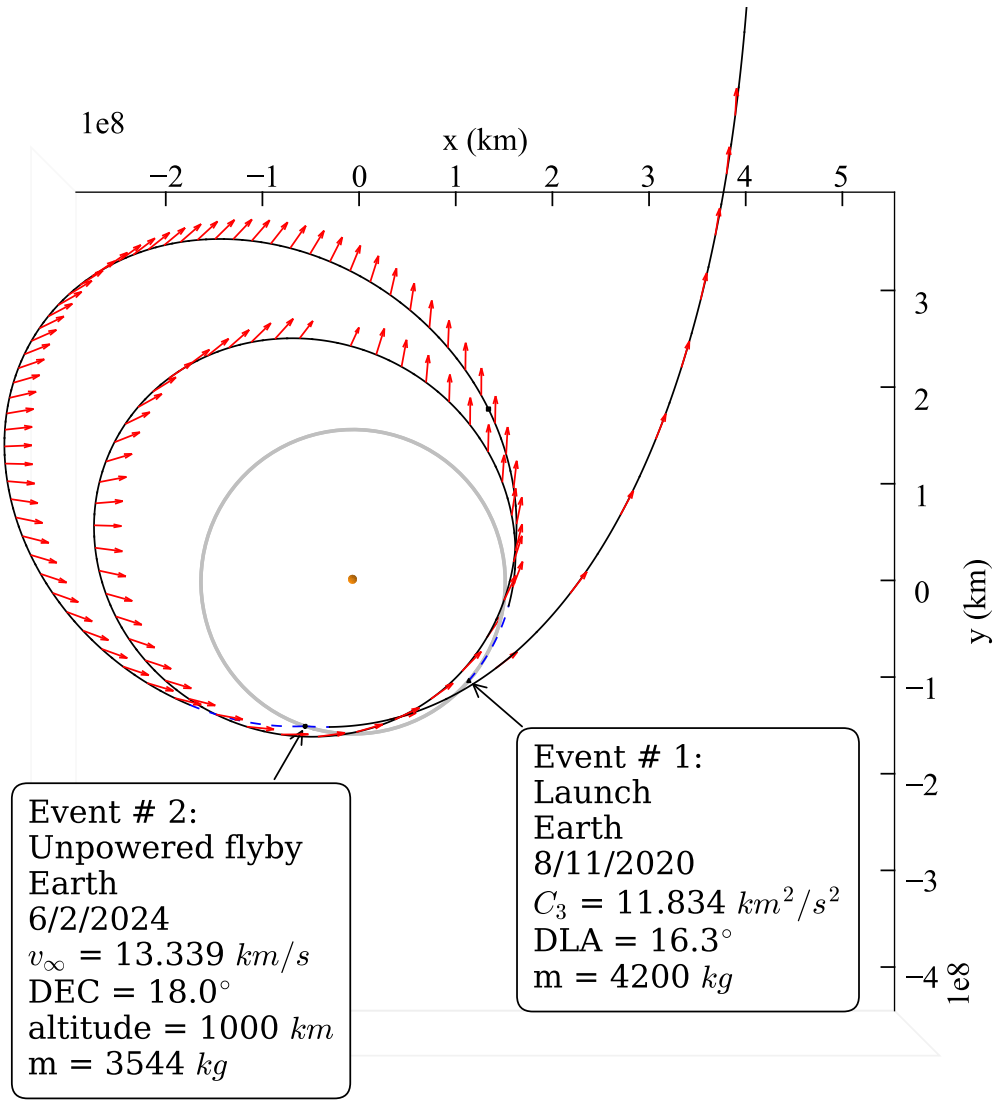

Figure 6. Optimal Uranus Probe trajectory.

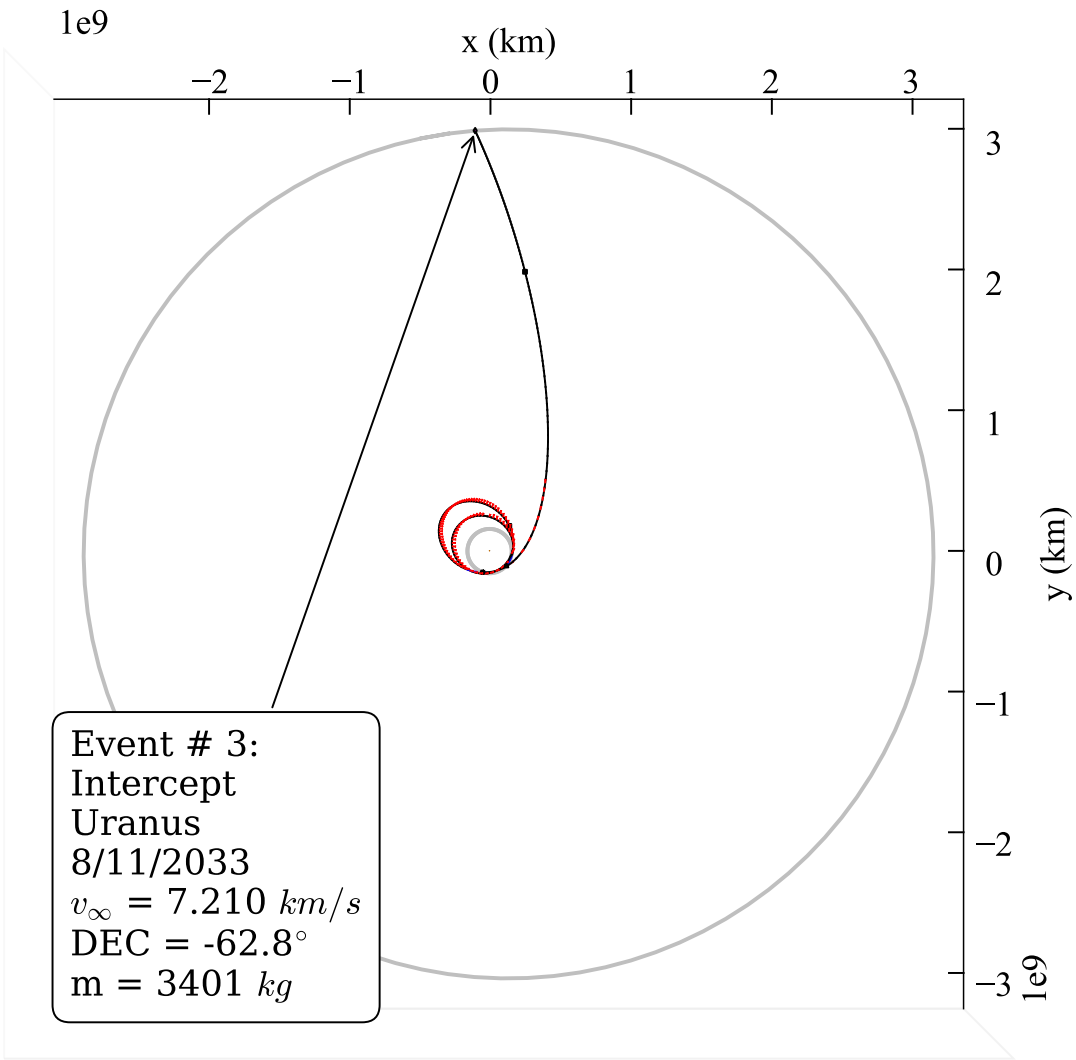

Figure 7. Optimal Uranus Probe trajectory.

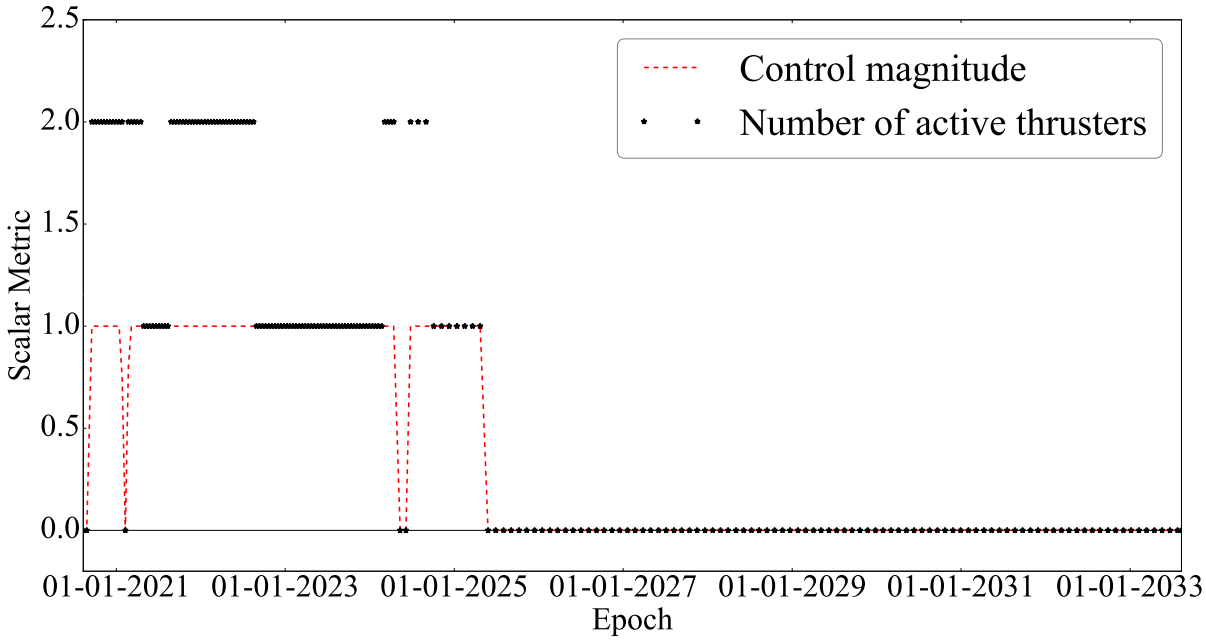

Figure 8. Optimal Uranus Probe trajectory throttle and active thruster histories.

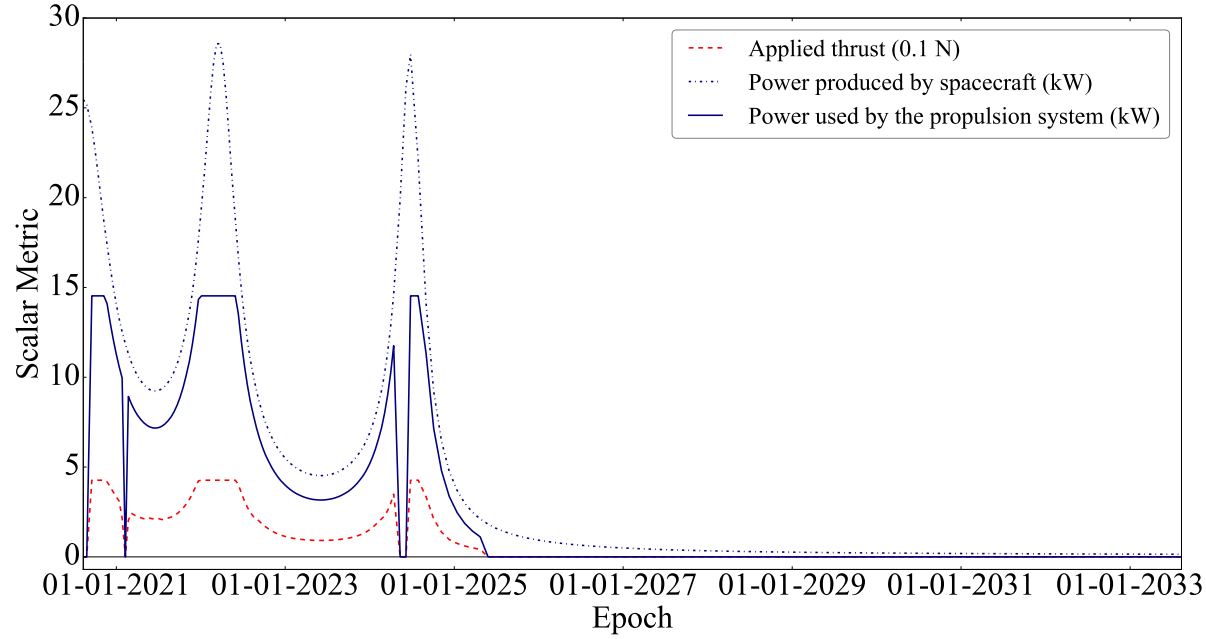

Figure 9. Optimal Uranus Probe trajectory throttle and active thruster histories.

V. B. Application to Mercury Rendezvous

For the next example, a more thorough investigation of the convergence properties of the NLP solver SNOPT was carried out. It should be noted that these results are specific to the the NLP solver used (SNOPT version 7.4-1.2), and the outcomes of these experiments are expected to vary slightly from solver to solver (and even between different versions of a solver). The mission considered is a low-thrust rendezvous with Mercury with a single intermediate flyby of Venus, and was previously solved by Sauer [21] and Debban et al. [22]. The objective function to be maximized is the mass delivered to Mercury. The problem was first solved using the MBH heuristic in conjunction with SNOPT, which provided the local search method. The problem assumptions are provided in Table 3 and the locally optimal solution that was found is shown in Figure 10 and has a normalized cost function of 0.701429.

Table 3. Mercury rendezvous mission parameters.

Parameter	Value
Maximum allowed initial mass	603 kg
Earliest allowed launch date	January 1st, 2002
Maximum flight time	3 years
Launch vehicle	Delta 7326
Launch C3	2.0 km^2/s^2
Launch declination bounds	$[-90.0°, 90.0°]$
Arrival type	rendezvous
Thruster	1 x NSTAR
Thruster duty cycle	1.0
Solar power coefficients	$\gamma_0 = 1.0$ $\gamma_1 = 0.0$ $\gamma_2 = 0.0$ $\gamma_3 = 0.0$ $\gamma_4 = 0.0$
Spacecraft bus power requirement coefficients	$a_{s/c} = 0.0$ $b_{s/c} = 0.0$ $c_{s/c} = 0.0$
Solar array BOL power (launch at 1 A.U.)	2.6 kW
Flyby sequence	E-V-Y
Minimum earth flyby altitude	300 km
Pre-Venus flyby coast	45 days
Post-Venus flyby coast	15 days
Number of segments per phase	200
Ephemeris	SplineEphem
SNOPT feasibility tolerance	1.0e-5
SNOPT optimality tolerance	1.0e-3
Objective function	maximize final mass
SNOPT max. runtime	1800 s
SNOPT major iteration limit	8000

Once the solution shown in Figure 10 was obtained, a series of convergence tests were performed by perturbing the optimal decision vector $\mathbf{x}^*$ with a series of random vectors drawn from a normal distribution $\mathcal{N}$ with a mean μ of zero:

$$x_i = x_i^* + \mathcal{N}(\mu = 0, \sigma_i) \;\; x_i \in \mathbf{x} \;\text{ and }\; \sigma_i \in \mathbb{R} \tag{12}$$

The standard deviation used to perturb the i^{th} decision variable (σ_i) was selected differently for each type of variable. A summary of these standard deviations is shown in Table 4.

The problem was then re-solved once using an analytic Jacobian and once with the partials provided by finite differences in order to draw a comparison between the two methods. This process was repeated for 100 trials. In each case, SNOPT was allowed to run for a maximum of 30 minutes. If the solver did not converge, exit stating infeasibility or otherwise terminate prior to the 30 minute limit, its achieved objective value was recorded as zero and its execution

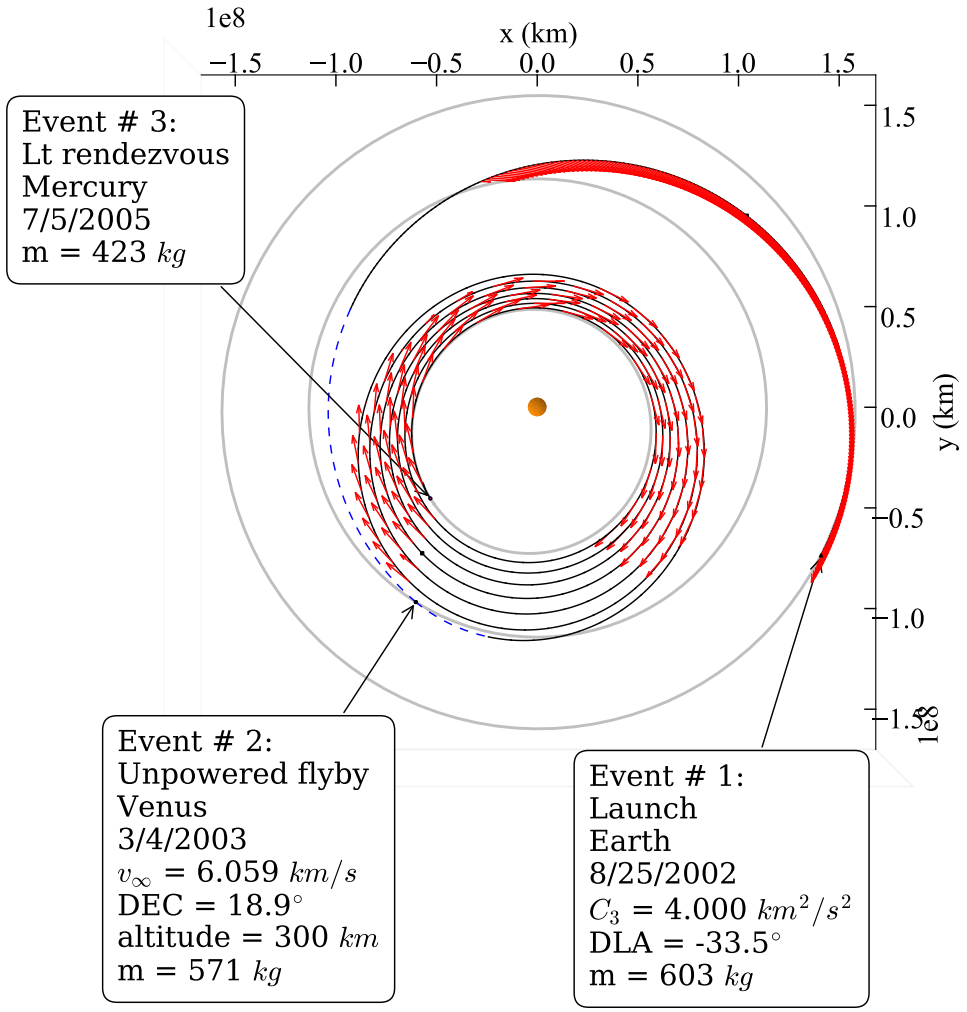

Figure 10. Optimal Mercury rendezvous trajectory.

Table 4. Standard deviations used to perturb each type of decision variable in the Mercury rendezvous problem.

Decision variable	σ_i
launch epoch	1.0 days
launch v_∞	1.0 km/s
right ascension of launch asymptote	0.5 radians
declination of launch asymptote	0.5 radians
Earth-Venus flight time	1.0 days
Earth-Venus terminal $\mathbf{v}_\infty$	1.0 km/s
Earth-Venus final mass	1.0 kg
Venus-Mercury flight time	1.0 days
Venus-Mercury initial $\mathbf{v}_\infty$	1.0 km/s
Venus-Mercury terminal $\mathbf{v}_\infty$	1.0 km/s
Venus-Mercury final mass	1.0 kg
control	0.1

time was recorded as 30 minutes. The objective function values achieved in each of these 200 trials are shown in Figure 11. The time-to-conclusion for each trial is shown in Figure 12.

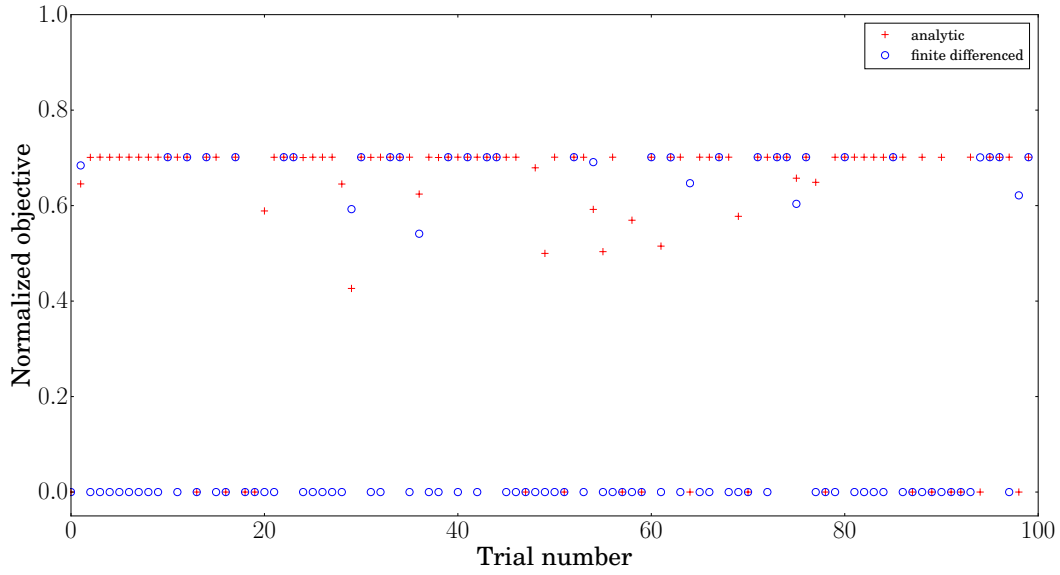

Figure 11. Final normalized objective values achieved by SNOPT for 100 perturbed initial conditions around the solution shown in Figure 10.

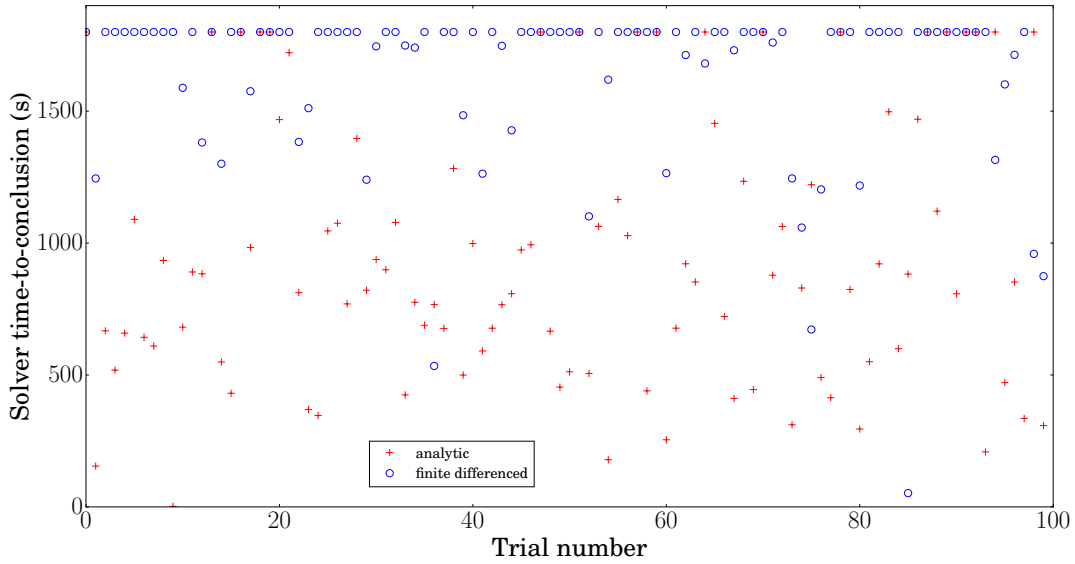

Figure 12. SNOPT time-to-conclusion for 100 perturbed initial conditions around the solution shown in Figure 10.

The times-to-conclusion in Figure 12 include instances where the solver successfully converged (feasibility and optimality tolerances satisfied), partially converged (feasible only), or successfully exited but did not converge (e.g. claimed problem was infeasible, encountered numerical difficulties during linear system decomposition etc.). Given a wall clock time limit of 30 minutes, using analytic derivatives SNOPT successfully reached a conclusion for 82/100

trials. Using finite differences it did so in 27/100 trials. Of these "successful conclusion" cases, 66 converged using an analytic Jacobian and 27 did so using finite differences. These statistics reveal an interesting finding. They suggest that when using analytic derivatives, not only is the solver able to converge more robustly, but also that its SQP algorithm is able to more effectively determine when it will *not* converge.

The next series of experiments that were performed sought to determine which decision variables were the most sensitive to perturbation for this problem, and how effective analytic derivatives are at re-converging a perturbed solution compared with finite differencing. The decision variables were divided into five categories: **time** (launch epoch, E-V flight time, V-Y flight time), **mass** (mass at V, mass at Y), **hyperbolic excess vectors** (launch v_∞, E-V final $\mathbf{v}_\infty$, V-Y initial $\mathbf{v}_\infty$, V-Y final $\mathbf{v}_\infty$), **control** and **launch asymptote orientation** (RA and DEC of launch asymptote). Using the standard deviations in Table 4, 20 perturbed initial decision vectors were created for each decision variable category. For example, for the **time** category only the launch epoch and phase flight time variables were perturbed; all other decision variables retained their optimal values. Each perturbed decision vector was used to initialize one SNOPT run once using analytic partials and one using finite differencing. A summary of convergence statistics for these experiments is provided in Table 5.

Table 5. Convergence rates for each decision variable category.

Decision variable category	Analytic convergence rate	Finite differencing convergence rate
time	19/20	16/20
mass	19/20	12/20
hyperbolic excess	16/20	5/20
control	19/20	11/20
launch RA/DEC	19/20	16/20

The results in Table 5 quantify the superiority of employing analytic gradients in the vicinity of an NLP solution. When employing an analytically defined Jacobian, SNOPT converged more frequently for all decision variable types.

V. C. Application to Odysseus Direct

The next mission design application considered is a flight to the "Greek camp" Jupiter Trojans that orbit the Sun-Jupiter L_4 point. Chemical missions to these asteroids invariably require a flyby of Jupiter (and possibly others to get to Jupiter) to facilitate the orbital phasing required to arrive at the L_4 point with a low enough v_∞ for a feasible rendezvous. With a continuous-thrust spacecraft, however, the most propellant-efficient way to get to the Trojans is a direct flight.

We consider a notional mission to 1184 Odysseus where a proximity constraint is placed on the spacecraft; it must not be farther than 5.75 AU from earth at any point during its transfer to Odysseus. This could be a communication range constraint for a low gain antenna to be used during the interplanetary leg of the mission. The value of 5.75 AU is somewhat arbitrary, however if the proximity constraint is reduced much more than that, the mass delivered to the asteroid is insufficient for a spacecraft bus powered with a 40 kW solar array and propulsion module containing supporting machinery for two NEXT thrusters. The mission parameters are provided in Table 6.

Table 6. Odysseus direct mission parameters

Parameter	Value
Maximum allowed initial mass	3500 kg
Earliest allowed launch date	January 1st, 2024
Launch window	365.25 days
Maximum flight time	14 years
Launch vehicle	Atlas V (551) NLS-2
Maximum launch C3	28.0 km^2/s^2
Launch declination bounds	$[-28.5^\circ, 28.5^\circ]$
Arrival type	low-thrust rendezvous
Thruster	2 x NEXT TT11, high thrust
Throttle logic	minimum number of thrusters
Thruster duty cycle	0.9
Solar power coefficients	$\gamma_0 = 1.0$ $\gamma_1 = 0.0$ $\gamma_2 = 0.0$ $\gamma_3 = 0.0$ $\gamma_4 = 0.0$
Spacecraft bus power requirement coefficients	$a_{s/c} = 1.0$ $b_{s/c} = 0.0$ $c_{s/c} = 0.0$
Solar array BOL power (launch at 1 A.U.)	40.0 kW
Power margin (δ_{power})	15%
Flyby sequence	E-O
Number of segments per phase	200
Ephemeris	SplineEphem
SNOPT feasibility tolerance	1.0e-5
Objective function	maximize final mass
MBH runtime	8 hrs.
SNOPT max. runtime (coarse FD analytic)	15 s 20 min
SNOPT major iteration limit	8000

One hundred MBH solver instances were run for this problem using the analytic methods introduced in this paper, using finite differencing and also using a "two-phase" approach where the two methods are combined. No initial guess was provided to the solver. This concept was first applied by Cassioli et al. [26] for solving interplanetary trajectory optimization problems using the MGA1DSM transcription [27]. The two-phase approach used for this problem begins each MBH search with SNOPT using finite differencing with a coarse step size ($h = 1.0e-2$) to compute the Jacobian. Once this coarse search has achieved a problem feasibility of 1.0e-3, the SQP method is continued from the same decision vector point $\mathbf{x}$ using the analytic methods described this work (fine search). For this fine search, SNOPT was allowed to run for a maximum of 20 minutes so that it might exploit the robustness that the derivatives provide and refine the solution as much as possible. The statistics for the 300 trials (100 for each Jacobian calculation strategy) are shown in Table 7 and the optimal solution found is shown in Figure 13.

Table 7. Solution statistics for 100 8-hour MBH runs.

Metric	Analytic	Finite differences	Two-phase
Best mass delivered (kg)	2123	2248	2295
Mean # feasible solutions	15	30	40
Mean-max mass delivered (kg)	1464	1719	2090
Mean-mean mass delivered (kg)	1143	1074	1977
Success rate (%)	98	99	100
Mean time to best solution	1 h 19 min.	40 min.	34 min.

The results in Table 7 indicate that, use of analytic gradients alone in the context of the hybrid MBH+SNOPT solver is not a good approach for this problem. Finite differencing performs better, however, by far the best strategy is the two-phase method. Analytic gradients increase the robustness of the SQP method, and the step direction determined by the minor level iterations is more likely to be accepted by the major level. When the NLP is not close to a feasible solution, and is perhaps not even in near the feasible region of the problem poor gradient information will result in

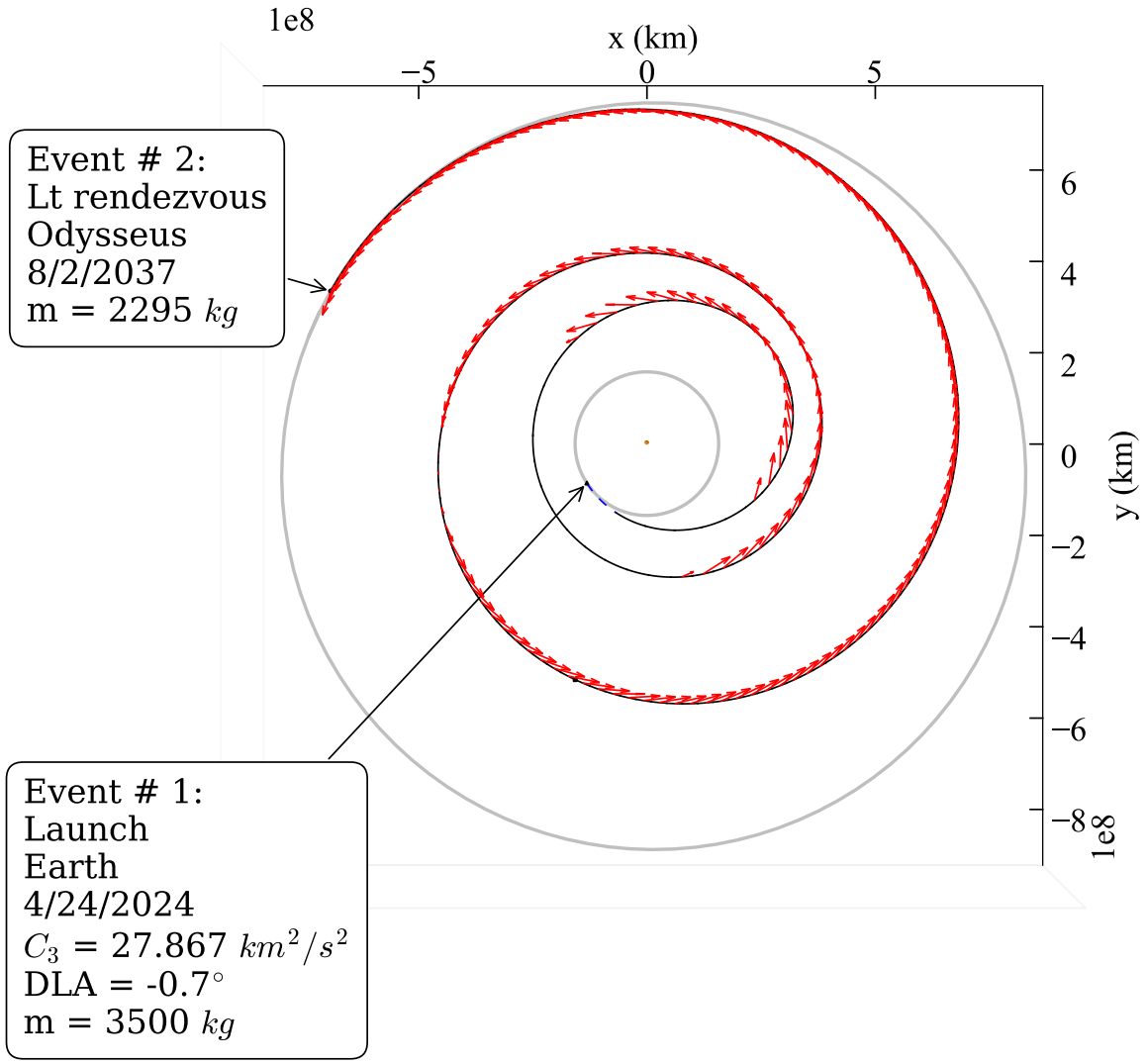

Figure 13. Optimal Odysseus Direct trajectory.

earlier termination of the SQP method. Analytic gradients, however, will provide accurate information to the SQP solver on how to improve the feasibility metric as well as the merit function, even if the algorithm ultimately will not converge. Therefore, since the MBH algorithm has a high probability of initializing SNOPT far from the feasible region, the analytic gradients can result in the SQP solver iterating in vain much of the time. As Cassioli et al. point out, gradients generated with a coarse step size will serve to smooth the objective function and the constraint space and postulated that the MGA1DSM problem possesses a funnel-like topology. The results shown here would suggest that the same might be true for the MGALT problem and that this technique benefits this transcription in the same way when basin hopping is employed. Once the SQP algorithm has progressed with the coarse search such that it has traversed the local funnel and the NLP is near-feasible, switching to analytic partials results in the solver exhibiting the robust convergence properties that were illustrated in the previous Mercury rendezvous example.

It should be noted that the success of the finite differencing algorithm at solving this problem, is in large part due

to the use of SplineEphem. When this problem was solved using SPICE (with finite differencing providing the partials of the ephemeris), the finite differencing runs failed to produce any feasible solutions, whereas the analytic partials fared about the same as when SplineEphem was used.

The same mission was also optimized without imposing the distance constraint. The optimal unconstrained trajectory is shown in Figure 14. Without the requirement that it remain within 5.75 AU of earth, the spacecraft passes well beyond the 5.75 AU limit prior to rendezvous with the asteroid. For this particular problem, the unconstrained and constrained problems are in the same family, and it can be concluded *a posteriori* that the proximity constraint could also have been enforced indirectly by constraining the arrival date at Odysseus.

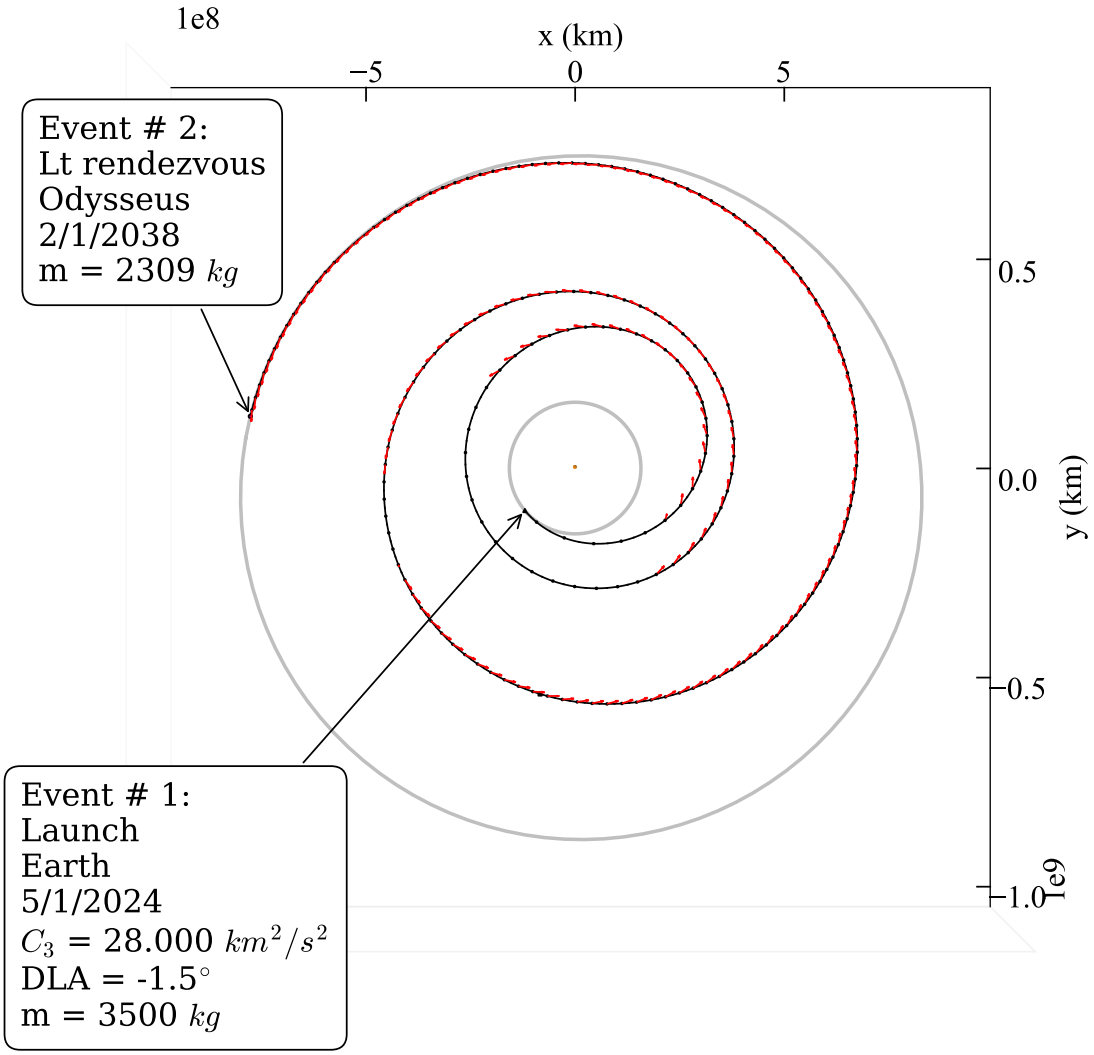

Figure 14. Optimal Odysseus Direct trajectory without proximity constraint.

V. D. Application to Cassini

The following example demonstrates the relative performance of analytic derivatives and finite differencing for the MGAnDSMs transcription. The Cassini problem solved here uses the same flyby sequence that the actual orbiter used on its way to Saturn (E-VVEJ-S). Due to the unavailability of a Titan IV (401) B launch vehicle model, it was assumed for this example that the spacecraft was launched using an Atlas V (551). The solver was allowed to place a single deep space maneuver at any point along each phase. An orbital insertion maneuver was included in the total mission ΔV and thus influenced the final delivered mass. The mission parameters and assumptions are described in Table 8. For this example problem SNOPT version 7.5 [28] was used. We found that with the more sensitive MGAnDSMs transcription, that SNOPT version 7.4-1.2 experienced significant numerical problems iterating at the SQP minor level quadratic sub-problem.

Table 8. Cassini mission parameters.

Parameter	Value
Earliest allowed launch date	January 1st, 1997
Saturn arrival date	unbounded
Launch window	365.25 days
Maximum flight time	7 years
Maximum allowed initial mass	10000 kg
Launch vehicle	Atlas V (551) NLS-2
Maximum launch C3	18.069 km^2/s^2
Launch declination bounds	$[-28.5^\circ, 28.5^\circ]$
Thruster I_{sp}	312 s
Flyby sequence	E-VVEJ-S
Ephemeris model	SPICE
Arrival type	orbital insertion
Insertion semimajor axis	5 447 500 km
Insertion eccentricity	0.98
SNOPT feasibility tolerance	1.0e-5
Max SNOPT runtime	5 s
Objective function	maximize $\log_{10}$ final mass

One hundred instances of the MBH solver were run for four hours. No initial guess was provided. All MBH runs converged to the known optimal solution. The best known optimal solution found delivers 3195 kg to orbit around Saturn, and requires only a single 0.428 km/s maneuver at aphelion between the two Venus flybys and a 0.622 km/s orbital insertion maneuver, totaling 1.05 km/s. This trajectory is shown in Figures 16 and 15.

Table 9. Solution statistics for 100 4-hour MBH runs.

Metric	Analytic	Finite Differences
Best mass delivered (kg)	3195	3195
Mean number of MBH hops taken	1432	591
Mean time to best solution	1 hr. 48 min.	2 hr. 10 min.

The relative performance between the runs using analytic derivatives and the those using finite differencing is more comparable for this transcription, as shown in Table 9, in part due to the fact that the Cassini problem is not particularly challenging for the MBH + SNOPT solver. As a result, both methods display considerable robustness, with none of the runs failing to identify the known optimal solution. The analytic derivatives arrived at their best found solution faster than the runs using finite differencing. This is due to the increased execution speed that employing analytic partials allows for in addition to the robustness benefits discussed in section V. B.

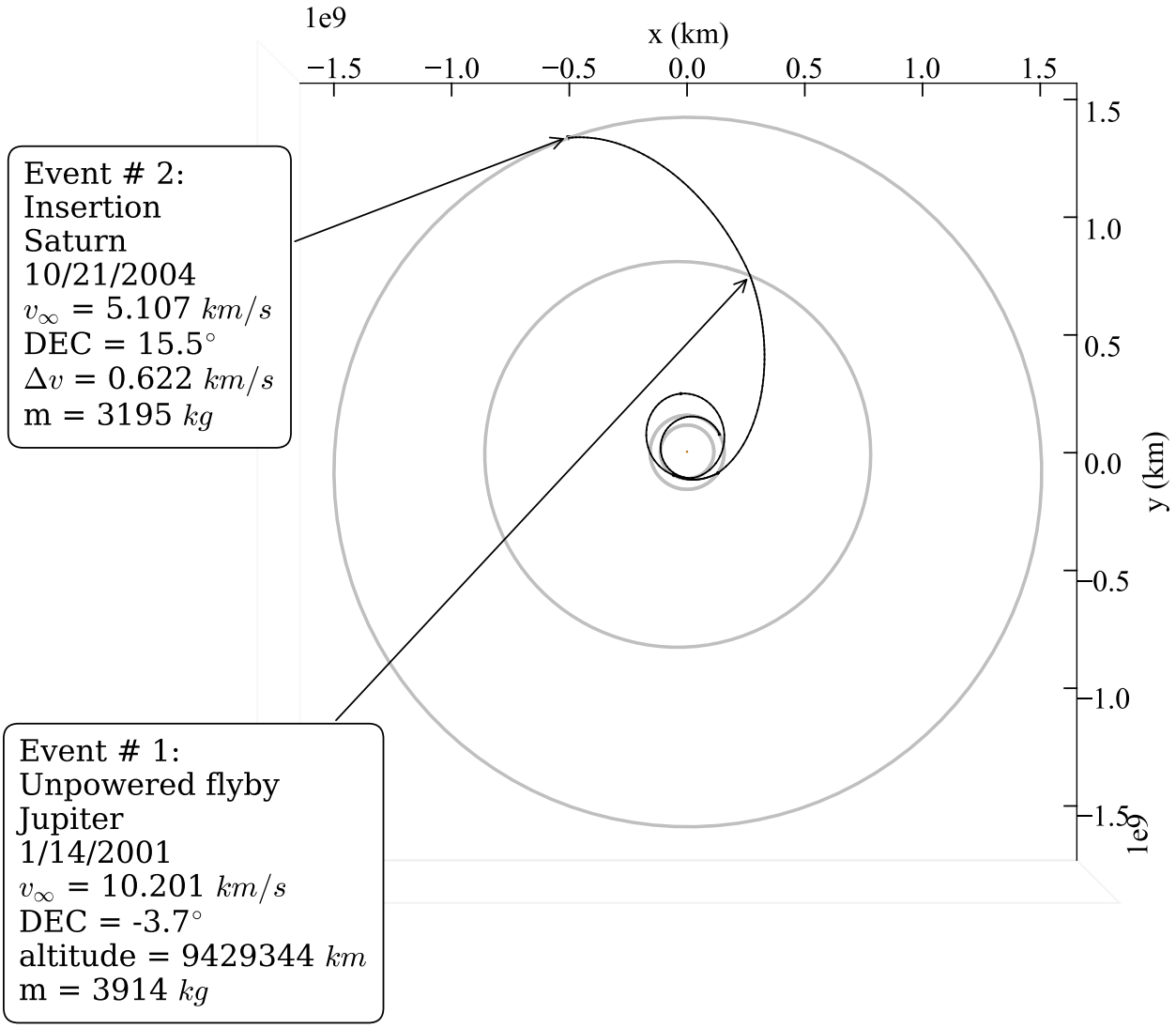

Figure 15. Optimal Cassini trajectory.

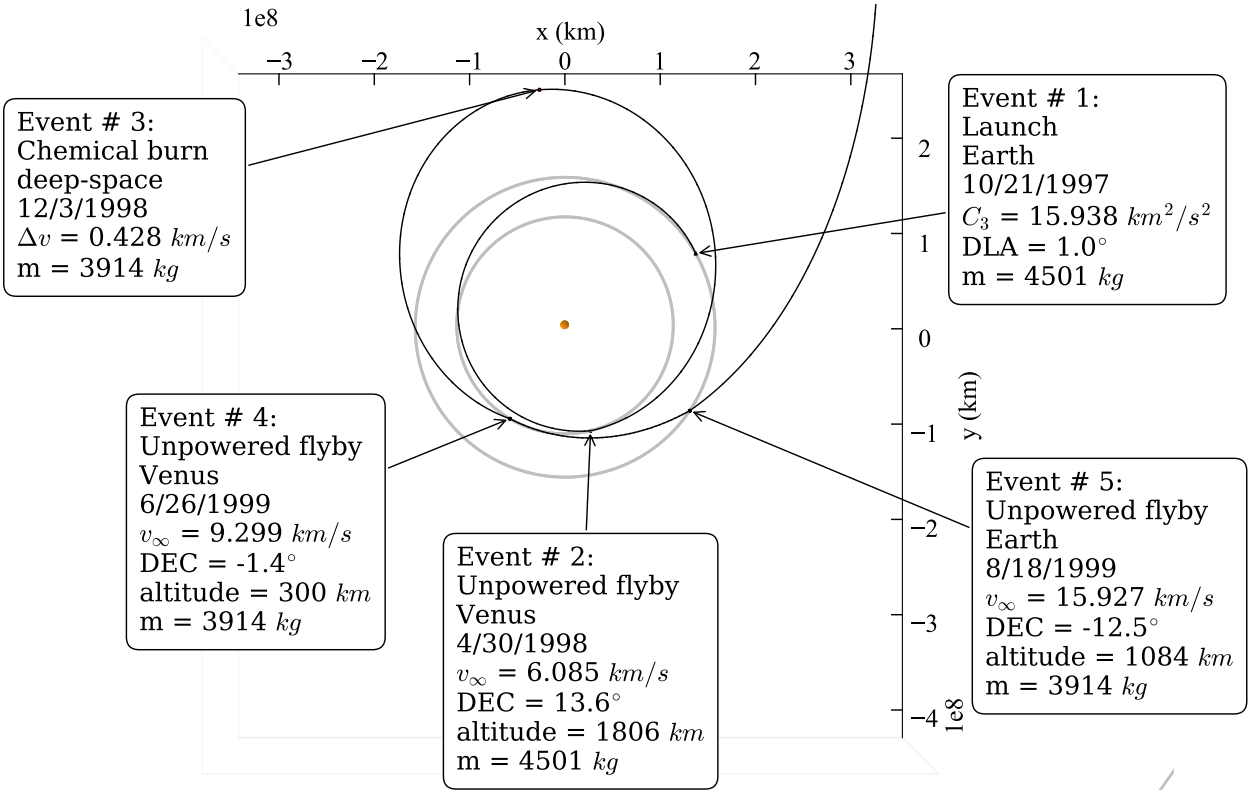

Figure 16. Optimal Cassini trajectory.

VI. Conclusion

The applications presented provide evidence in favor of using analytic gradients over ones computed using the method of finite differences, when possible. Their use is especially attractive in the context of a hybrid optimal control solver structure that requires the execution of a large number of local searches. The monotonic basin hopping heuristic method benefits considerably when it is coupled with a nonlinear programming solver such as SNOPT using these techniques. A greater improvement from using analytic expressions for the Jacobian entries vs. computing them with finite differences was seen for the continuous-thrust (MGALT) model than for the impulsive thrust (MGAnDSMs) model. The example problems solved indicate that as the NLP cost function complexity increases, the improvements in speed and robustness are more pronounced. These examples also underscore the importance of incorporating an accurate power system model during the preliminary phase of a mission concept study. One such power model, was presented that features a solar electric array model as well as a method for incorporating discrete numbers of active thrusters by way of a smoothing technique based on the Logistics function. This model is compatible with the theory developed in the companion paper.

Appendix

A. Thruster Performance Curves

Thrust polynomial coefficients provide a relationship between input power (kW) and the maximum possible applied thrust (mN). Mass flow rate coefficients provide a curve fit in mg/s.

Table 10. Thruster performance parameters.

Parameter	NEXT TT10 High-I_{sp}	NEXT TT11 [a] High-Thrust	NSTAR [b]
a_T	-1.92224e-6	11.9388817	26.337459
b_T	54.05382	16.0989424	-51.694393
c_T	-14.41789	11.4181412	90.486509
d_T	2.96519	-2.04053417	-36.720293
e_T	-0.19082	0.101855017	5.145602
a_m	2.13781	2.75956482	2.5060
b_m	0.03211	-1.71102132	-5.3568
c_m	-0.09956	1.21670237	6.2539
d_m	0.05717	-0.207253445	-2.5372
e_m	-0.004776	0.011021367	0.36985
P_{max} (kW)	7.266	7.36	2.6
P_{min} (kW)	0.638	0.64	0.525

B. Throttle Mode Logic

The following Algorithms describe two of the multi-thruster logic programs described in section III.

Algorithm 1 Multi-Thruster Logic:
Maximum Number of Thrusters

$n_{active} = n_{available}$
if $P \geq P_{min}$ **then**
  for $n = n_{available}; n > 0; --n$ **do**
    if $P \geq nP_{min}$ **then**
      $n_{active} = n$
      break
    end if
  end for
else
  $n_{active} = 0$
end if

Algorithm 2 Multi-Thruster Logic:
Minimum Number of Thrusters

$n_{\text{active}} = n_{\text{available}}$
if $P \geq P_{\text{min}}$ **then**
  for $n = n_{\text{available}}; n > 0; --n$ **do**
    if $P \leq nP_{\text{max}}$ **then**
      $n_{\text{active}} = n$
      break
    end if
  end for
else
  $n_{\text{active}} = 0$
end if

C. Monotonic Basin Hopping

Algorithm 3 Monotonic Basin Hopping (MBH)

generate random point $\mathbf{x}$

run NLP solver to find point $\mathbf{x}^*$ using initial guess $\mathbf{x}$

$\mathbf{x}_{\text{current}} = \mathbf{x}^*$

if $\mathbf{x}^*$ is a feasible point **then**

  save $\mathbf{x}^*$ to archive

end if

while not hit stop criterion **do**

  generate $\mathbf{x}'$ by randomly perturbing $\mathbf{x}_{\text{current}}$

  for each time of flight variable t_i in $\mathbf{x}'$ **do**

    if $rand\,(0,1) < \rho_{\text{time-hop}}$ **then**

      shift t_i forward or backward one synodic period

    end if

  end for

  run NLP solver to find locally optimal point $\mathbf{x}^*$ from $\mathbf{x}'$

  if $\mathbf{x}^*$ is feasible **and** $f\,(\mathbf{x}^*) < f\,(\mathbf{x}_{\text{current}})$ **then**

    $\mathbf{x}_{\text{current}} = \mathbf{x}^*$

    save $\mathbf{x}^*$ to archive

  else if $\mathbf{x}^*$ is infeasible **and** $\|\mathbf{c}\,(\mathbf{x}^*)\| < \|\mathbf{c}\,(\mathbf{x}_{\text{current}})\|$)

    $\mathbf{x}_{\text{current}} = \mathbf{x}^*$

  end if

end while

return best $\mathbf{x}^*$ in archive

In Algorithm 3, $\rho_{\text{time-hop}}$ is the probability of advancing the state of the spacecraft by one synodic period as described by Cassioli et al. [26].

D. Example MGALT STM-MTM Chain Multiplication

It is useful, for implementation verification purposes, to compute a small chain multiplication of STM and MTM matrices. Equation (13) contains the numerical results of computing an STM-MTM chain consisting of two maneuvers, using the parameters shown in Table 11. The initial position $\mathbf{r}_0$ and velocity $\mathbf{v}_{\text{ephem}}$ were generated using SPICE N0065 for earth (SPICE ID 399) at the reference epoch t_0. An initial impulse $\mathbf{v}_\infty$ was then added to form $\mathbf{v}_0$. It should be noted that this example propagation does not include any thruster activation/deactivation events:

Table 11. Parameters for STM-MTM example multiplication

Parameter	Value
$\mathbf{v}_{\text{ephem}}$	$\begin{bmatrix} -17.50086332293890 \\ 22.31903051279331 \\ 9.676131675525272 \end{bmatrix}$ km/s
$\mathbf{v}_\infty$	$\begin{bmatrix} 3.554492889702753 \\ 5.179827707539178 \\ 3.019249114806855 \end{bmatrix}$ km/s
$\mathbf{r}_0$	$\begin{bmatrix} 121954619.5281144 \\ 77941136.13335293 \\ 33789204.90478533 \end{bmatrix}$ km
$\mathbf{v}_0$	$\begin{bmatrix} -13.94637043323615 \\ 27.49885822033249 \\ 12.69538079033212 \end{bmatrix}$ km/s
$\mathbf{u}_1^T$	$\begin{bmatrix} 0.55013287953737899 \\ 0.67720326929228092 \\ 0.48862004707126649 \end{bmatrix}$
$\mathbf{u}_2^T$	$\begin{bmatrix} 0.54375105665486845 \\ -0.42138762125764850 \\ 0.52883862353025801 \end{bmatrix}$
m_0	525.2 kg
γ_0	1.32077
γ_1	-0.10848
γ_2	-0.11665
γ_3	0.10843
γ_4	-0.012790
$\{a_{s/c}, b_{s/c}, c_{s/c}\}$	0.0
P_0	2.0 kW at 1 A.U.
t_0	246869420.3576459 s past J2000 (MJD 54401.78495784312)
Δt_p	302.0 days
Δt_1	652320.00000000 s
$\Delta t_2 = \Delta t_3$	1304640.0000000 s

$$\mathbf{\Phi}(t_3, t_0) = \mathbf{\Phi}_3 \mathbf{M}_2 \mathbf{\Phi}_2 \mathbf{M}_1 \mathbf{\Phi}_1 \tag{13}$$

$$\tilde{\mathbf{R}}(t_3, t_0) = \begin{bmatrix} 1.002799475749591 & 0.2875110052470521 & 0.1276999938149953 \\ 0.3072394653658856 & 1.18690609143207 & 0.1861633207764594 \\ 0.1347994753703645 & 0.1842310463476051 & 0.851004072508968 \end{bmatrix} \tag{14}$$

$$\mathbf{R}(t_3, t_0) = \begin{bmatrix} 3821756.624139383 & 297901.6090727107 & 133186.6106473491 \\ 312038.7623979754 & 4199579.456851517 & 253071.1699968702 \\ 136121.8522236888 & 248759.3937067411 & 3746641.127550326 \end{bmatrix} \tag{15}$$

$$\tilde{\mathbf{V}}(t_3, t_0) = \begin{bmatrix} -2.277307825912253e-08 & 1.148113971328598e-07 & 5.118699510664434e-08 \\ 1.364234207336634e-07 & 1.200483077722063e-07 & 1.003820877882617e-07 \\ 5.990774736286879e-08 & 9.92837118077669e-08 & -5.98205955166005e-08 \end{bmatrix} \tag{16}$$

$$\mathbf{V}(t_3, t_0) = \begin{bmatrix} 0.8933499554785707 & 0.1686875608843946 & 0.07575948583351597 \\ 0.1860440560176314 & 1.245483220921972 & 0.1972895343860645 \\ 0.08138975782445479 & 0.1944526712193977 & 0.8952504811455283 \end{bmatrix} \tag{17}$$

$$\frac{\partial m_3}{\partial \mathbf{r}_0} = \begin{bmatrix} 4.14218882637117e-08 & 4.43666780306202e-08 & 1.962244271863216e-08 \end{bmatrix} \tag{18}$$

$$\frac{\partial m_3}{\partial \mathbf{v}_0} = \begin{bmatrix} 0.06832824048184363 & 0.08066882389471471 & 0.03577809533022297 \end{bmatrix} \tag{19}$$

$$\frac{\partial \mathbf{r}_3}{\partial m_0} = \begin{bmatrix} -707.2477287869197 \\ -473.033434430855 \\ -652.312461085274 \end{bmatrix} \tag{20}$$

$$\frac{\partial \mathbf{v}_3}{\partial m_0} = \begin{bmatrix} -0.0003531999144470822 \\ -0.0001513603072789627 \\ -0.0003391022873649247 \end{bmatrix} \tag{21}$$

$$\frac{\partial \mathbf{r}_3}{\partial \Delta t_p} = \begin{bmatrix} -3.765869764196434 \\ 1.588865507941996 \\ 0.7992696543032576 \end{bmatrix} \tag{22}$$

$$\frac{\partial \mathbf{v}_3}{\partial \Delta t_p} = \begin{bmatrix} -7.356819085428689e-08 \\ -5.303853530010765e-07 \\ -2.358899494817949e-07 \end{bmatrix} \tag{23}$$

$$\frac{\partial m_3}{\partial \Delta t_p} = -2.05961273325598e-07 \tag{24}$$

Acknowledgments

The authors would like to thank Matthew Vavrina for his work developing the MGAnDSMs transcription and Dr. Alexander Ghosh at the University of Illinois for the use of his automatic differentiation library. In addition, Dr. Englander would like to thank the internal research and development (IRAD) program and the planetary science new business office at NASA Goddard Space Flight Center and Dr. Ozimek would like to thank the Applied Physics Laboratory Civilian Space IR&D program. This work made use of the Illinois Campus Cluster, a computing resource that is operated by the Illinois Campus Cluster Program (ICCP) in conjunction with the National Center for Supercomputing Applications (NCSA) and which is supported by funds from the University of Illinois at Urbana-Champaign.

References

[1] Prussing, J. E. and Chiu, J. H., "Optimal Multiple-Impulse Time-Fixed Rendezvous Between Circular Orbits," *Journal of Guidance, Control, and Dynamics*, Vol. 9, No. 1, 1986, pp. 17–22, doi:10.2514/3.20060.

[2] Sims, J. A. and Flanagan, S. N., "Preliminary Design of Low-Thrust Interplanetary Missions," in "AAS/AIAA Astrodynamics Specialist Conference, AAS Paper 99-338," Girdwood, Alaska, 1999.

[3] Sims, J., Finlayson, P., Rinderle, E., Vavrina, M., and Kowalkowski, T., "Implementation of a Low-Thrust Trajectory Optimization Algorithm for Preliminary Design," in "AIAA/AAS Astrodynamics Specialist Conference, AIAA Paper 2006-6746," , 2006.

[4] Yam, C., di Lorenzo, D., and Izzo, D., "Low-Thrust Trajectory Design as a Constrained Global Optimization Problem," in "Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering," Vol. 225, 2011, pp. 1243–1251.

[5] Englander, J., Conway, B., and Williams, T., "Automated Mission Planning via Evolutionary Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 6, 2012, pp. 1878–1887, doi:10.2514/1.54101.

[6] Vavrina, M., Englander, J. A., and Ghosh, A. R. M., "Coupled Low-Thrust Trajectory and Systems Optimization via Multi-Objective Hybrid Optimal Control," in "AAS/AIAA Space Flight Mechanics Meeting, AAS Paper 15-397, Williamsburg, VA," , 2015.

[7] Englander, J. A., Vavrina, M., and Ghosh, A. R. M., "Multi-Objective Hybrid Optimal Control for Multiple-Flyby Low-Thrust Mission Design," in "AAS/AIAA Space Flight Mechanics Meeting, AAS Paper 15-227, Williamsburg, VA," , 2015.

[8] Ellison, D. H., Conway, B. A., Englander, J. A., and Ozimek, M. T., "Partial Derivative Computation for Bounded-Impulse Trajectory Models Using Two-Sided Direct Shooting. Part 1: Theory," *Journal of Guidance, Control, and Dynamics*, Vol. XX, No. XX, XXXX, pp. XXX–XXX.

[9] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Rev.*, Vol. 47, No. 1, 2005, pp. 99–131, doi:10.1137/S0036144504446096.

[10] Wächter, A. and Biegler, L. T., "On the Implementation of a Primal-Dual Interior Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, Vol. 106, No. 1, 2006, pp. 25–57, doi:10.1007/s10107-004-0559-y.

[11] Leary, R., "Global Optimization on Funneling Landscapes," *Journal of Global Optimization*, Vol. 18, No. 4, 2000, pp. 367–383, doi:10.1023/A:1026500301312.

[12] Vasile, M., Minisci, E., and Locatelli, M., "Analysis of Some Global Optimization Algorithms for Space Trajectory Design," *Journal of Spacecraft and Rockets*, Vol. 47, No. 2, 2010, pp. 334–344, doi:10.2514/1.45742.

[13] Englander, J. A., *Automated Trajectory Planning for Multiple-Flyby Interplanetary Missions*, Ph.D. thesis, University of Illinois at Urbana-Champaign, 2013.

[14] Englander, J. A. and Englander, A. C., "Tuning Monotonic Basin Hopping: Improving the Efficiency of Stochastic Search as Applied to Low-Thrust Trajectory Optimization," in "24th International Symposium on Space Flight Dynamics, Laurel, MD," , 2014.

[15] Vavrina, M. and Howell, K., "Global Low Thrust Trajectory Optimization through Hybridization of a Genetic Algorithm and a Direct Method," in "AIAA/AAS Astrodynamics Specialist Conference and Exhibit, AIAA Paper 2008-6614," Honolulu, Hawaii, 2008, doi:10.2514/6.2008-6614.

[16] McConaghy, T. T., *Design and optimization of interplanetary spacecraft trajectories*, PhD dissertation, School of Aeronautics and Astronautics, Purdue University, 2004.

[17] Bracewell, R., ed., *The Fourier Transform & Its Applications, 3rd ed.*, McGraw-Hill Science/Engineering/Math, 2000.

[18] "SPICE Ephemeris," , 2013. http://naif.jpl.nasa.gov/naif/.

[19] Arora, N. and Russell, R. P., "A fast, accurate, and smooth planetary ephemeris retrieval system," *Celestial Mechanics and Dynamical Astronomy*, Vol. 108, No. 2, 2010, pp. 107–124, doi:10.1007/s10569-010-9296-0.

[20] Galassi, M., *GNU Scientific Library Reference Manual (3rd edition).*

[21] Sauer Jr., C. G., "AAS 97-726 Solar Electric Performance for Medlite and Delta Class Planetary Missions," in "AAS/AIAA Astrodynamics Specialist Conference," Sun Valley, Id, 1997.

[22] Debban, T. J., McConaghy, T. T., and Longuski, J. M., "AIAA 2002-4729 Design and Optimization of Low-Thrust Gravity-Assist Trajectories to Selected Planets," in "AIAA/AAS Astrodynamics Specialist Conference and Exhibit," Monterey, CA, August, 2002, pp. 1–10,
doi:10.2514/6.2002-4729.

[23] Hubbard, W. B., "Ice Giants Decadal Study," Tech. rep., NASA Planetary Sciences Decadal Survey, 2010. http://sites.nationalacademies.org/SSB/SSB_059331.

[24] "NASAs Evolutionary Xenon Thruster (NEXT) Ion Propulsion GFE Component Information Summary for Discovery Missions July 2014," , 2014. http://discovery.larc.nasa.gov/discovery/pdf_files/20-NEXT-C_AO_Guidebook_11July14.pdf.

[25] Council, N. R., *Vision and Voyages for Planetary Science in the Decade 2013-2022*, The National Academies Press, Washington, DC, 2011,
doi:10.17226/13117.

[26] Cassioli, A., Izzo, D., Di Lorenzo, D., Locatelli, M., and Schoen, F., *Global Optimization Approaches for Optimal Trajectory Planning*, Springer, chap. 5, pp. 111–140, 2013,
doi:0.1007/978-1-4614-4469-5_5.

[27] Izzo, D., *Spacecraft Trajectory Optimization*, Cambridge University Press, chap. 7, pp. 178–200, 2010.

[28] Gill, P. E., Murray, W., Saunders, M. A., and Wong, E., "User's Guide for SNOPT 7.5: Software for Large-Scale Nonlinear Programming," Center for Computational Mathematics Report CCoM 15-3, Department of Mathematics, University of California, San Diego, La Jolla, CA, 2015.

[29] Oh, D. and Goebel, D., "Performance evaluation of an expanded range XIPS ion thruster system for NASA science missions," in "42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, AIAA Paper 2006-4466, Sacramento, CA," , 2006,
doi:10.2514/6.2006-4466.